sainho 02月14日

第七章 | OmniFocus 与其他工具的协作

| 本文为付费栏目文章，您已订阅，可阅读全文 |

每当我折腾 Drafts、Launch Center Pro 这种效率工具，或者 Airmail 等支持第三方应用协作的应用时，我几乎是下意识地认为：

它一定能和 OmniFocus 协作！

自发布以来，OmniFocus 已经有快 10 年的历史。这期间不仅紧跟 iOS 和 macOS 的新系统特性，也开放了丰富的应用接口，使得其他应用能够与 OmniFocus 进行协作。与此同时，OminFocus 自身的优秀和风靡，也吸引了不少第三方开发者愿

意主动与适配 OmniFocus。所以我才敢有些「武断」地作出如上判断。

前六章我们基本上已经探索了 OmniFocus 的方方面面。在这教程的最后一章，我们就来看看 OmniFocus 如何与 iOS & macOS 上的其他应用进行协作。

OmniFocus 为什么要和其他应用进行协作？

我们先来明确一个「残酷」的事实：**所谓的「协作」并不是万能的，和其他应用进行协作，并不能为 OmniFocus 增加新功能。**

我们在第一章中举过 Launch Center Pro 的例子，知道如何通过 Launch Center Pro 在 iOS 中添加 OmniFocus 动作。但 OmniFocus 只能为动作或者项目添加一个上下文，无论是多么厉害的第三方应用，都不可能改变这个功能设计。此外由于 iOS 上的沙盒机制，使得在 iOS 上，那些已经被 OmniFocus 被保存的任务，不可能像在 Mac 上一样，再被外部工具修改。

看起来限制这么多，所谓的「协作」到底能到些什么？又能给我们带来什么样的好处呢？

尽管在 iOS 上应用之间被沙盒机制隔离开来，但我们还是可以利用 URL Schemes 来传递数据。iOS 上第三方应用能为 OmniFocus 做的，就是**尽可能地丰富添加动作的场景**。有不少喜爱使用 Drafts 的用户，会将 Drafts 作为唯一记录碎片文字的地方。若是走在学校路上看到秋招会的信息，打开 Drafts 就记

下了时间地点。之后回顾时想创建一个「制作秋招会需要的简历」的动作，如果能从 Drafts 里，利用之前的笔记直接创建，无疑会方便的多。

而对于 OmniFocus Mac 版来说，系统带来的限制要小得多，AppleScript 能做的事也比 URL Schemes 要多。所以我们所期待的，是能和 Evernote 这类我们习惯使用的应用，组成一套工作流。或者利用 Alfred 等工具，进一步提升使用时的操作效率。

总而言之，**应用间的协作是用来方便「懒人」的。**本来需要我们手动在应用间跳来跳去的操作，通过「协作」能变得「自动化」。

OmniFocus 在 iOS 中与其他应用协作

因为沙盒机制在 iOS 中的存在， URL Schemes 对于应用间协作就变得举足轻重。OmniFocus 中的 URL Schemes 的变化，影响着它与其他应用的协作的能力。

如果你还对 URL Schemes 不了解，可以先阅读 Hum 的《URL Schemes 使用详解》。

早期的 OmniFocus，只能支持一些简单导航类型的 UR
Schemes，方便用户直接打开某个透视或者项目。例如
URL Schemes：

- `omnifocus:///inbox` ：显示收件箱

- `omnifocus:///folder/文件夹 ID` ：显示具体文件夹
- `omnifocus:///task/项目 ID` ：显示具体项目
- `omnifocus:///context/情境 ID` ：显示具体情境
- `omnifocus:///perspective/透视名` ：显示具体透视
- `omnifocus:///review` ：显示检查
- `omnifocus:///nearby` ：显示附近任务
- `omnifocus:///flagged` ：显示旗标任务
- `omnifocus:///past` ：显示截止日期为过去的任务
- `omnifocus:///today` ：显示截止日期为今天的任务
- `omnifocus:///soon` ：显示截止日期为未来的任务

最关键的添加任务的动作 `/add` ，也只有动作名和附注可以添加，所以只能写出类似这种程度的 URL Schemes：

```
omnifocus:///add?name=买两瓶牛奶&note=要全脂的
```

在这种情况下，我认为几乎是没有什么和第三方应用协作的必要性的。例如「买两瓶牛奶」这个动作，既没法添加上下文也没法添加位置信息，都要打开应用再次手动添加，实在方便不到哪里去。而导航类的 URL Schemes，也只是方便 LCP 这类[illegible]应用打开而已。

2016 年 4 月的发布的 OmniFocus for iOS 2.14 版本中，发生了两个可谓关键的改变：

1. 丰富了 `/add` 动作所支持的参数，基本涵盖了动作和项目的关键属性；
2. 支持 TaskPaper 格式的文本。

直接通过 URL Schemes 添加动作

首先值得关注的是第一项改变，在 2.14 版本中新增加了下来 `/add` 动作的参数：

- `name`：动作标题、任务名；
- `note`：动作附注；
- `estimate`：预估的延续时间，例如 `2h` 表示两小时，`3w` 表示三周；
- `flag` ：标为旗帜（true 或者 false）；
- `parallel` ：项目类型为平行（true）或顺序（false）；
- `autocomplete`：自动完成（true 或者 false），当动作完成时子项是否自动标为完成；
- `defer` ：推迟至日期，例如 `2016-04-19 5pm` ；
- `due` ：截止日期；
- `completed` ：完成日期；
- `repeat-method` ：重复方式，`fixed` 定期重复，`start-after-completion` 完成后再次推迟，`due-after-completion` 完成后再次到期；

- `repeat-rule` ：重复规则，填写的内容要遵从 ICS 的语法，文档见 RFC2445。例如 `FREQ=WEEKLY;INTERVAL=1` （重复周期为一周，次数为一次）；
- `project` ：项目名；
- `context` ：上下文；
- `attachment-name` ：附件名；
- `attachment` ：附件；
- `reveal-new-item` ：创建动作后是是否会跳转到动作所在页面（true 或者 false）。

这么齐全的属性参数，让通过 URL Scheme 添加动作不再鸡肋。对于那些习惯使用 Launch Center Pro 的人来说，完全可以在 Launch Center Pro 里填写所有属性，不必先点开 OmniFocus，再一个个选项去选择。在第一章中我们已经介绍过如何使用，这里不再赘述。只提醒一句，在第一章中当时为了简化流程，并没有将全部的属性都派上用场。实际使用时上面列举的属性，都可以通过 Launch Center Pro 添加。

通过 TaskPaper 格式文本添加动作

通过 Launch Center Pro 添加动作，所支持的属性非常完善，但有个不可避免的弊病：在 Launch Center Pro 中的动作 一旦创建就不会每次去手动修改。为了节约输入时的时间，[illegible]想不会有人会在创建时选中所有的参数，例如 `repeat-rule` 这类不常用的参数就不会选择。但一旦需要时，要么修改 Launch

Center Pro 的动作，要么进了 OmniFocus 再去手动添加，两个方案都比原先要多出一步操作来。

2.14 版本更新中，OmniFocus 支持导入和导出 TaskPaper 格式文本，让我们有一种易读好写又灵活的方式来添加动作，避免 Launch Center Pro 的弊端。

TaskPaper 是一款纯文本的任务管理软件。不同于 OmniFocus 都是通过图形界面来设置动作属性，TaskPaper 通过它所定义的格式来给任务设置属性。TaskPaper 格式非常易读，例如：

```
琐事:
交房租 @parallel(true) @autodone(false) @flagged @context(Timing : Low Power) @due(2018-03-03 17:00) @defer(2018-03-01 00:00) @repeat-method(fixed) @repeat-rule(FREQ=MONTHLY)
```

就表示**平行项目**「琐事」中，有一个**推迟至** 2018 年 1 月 1 日，**截止日**是 1 月 3 日， 每月**定期重复**，**上下文**是「Low Power」的**已标注**动作。

和 OmniFocus 对比，会发现 TaskPaper 格式文本中，项目名后跟一个 `:` ，动作名前是 `-` ，其余的属性全部前置一个 `@` 即可。而如果要把这个动作写成 URL Schemes 的形式，就是：

```
omnifocus:///add?name=交房租&autocomplete=false&repeat-method=fixed&repeat-rule=FREQ=MONTHLY&parallel=tru
```

```
e&flag=true&defer=2018-03-01%205pm&due=2018-03-03%20 00:00&project=琐事
```

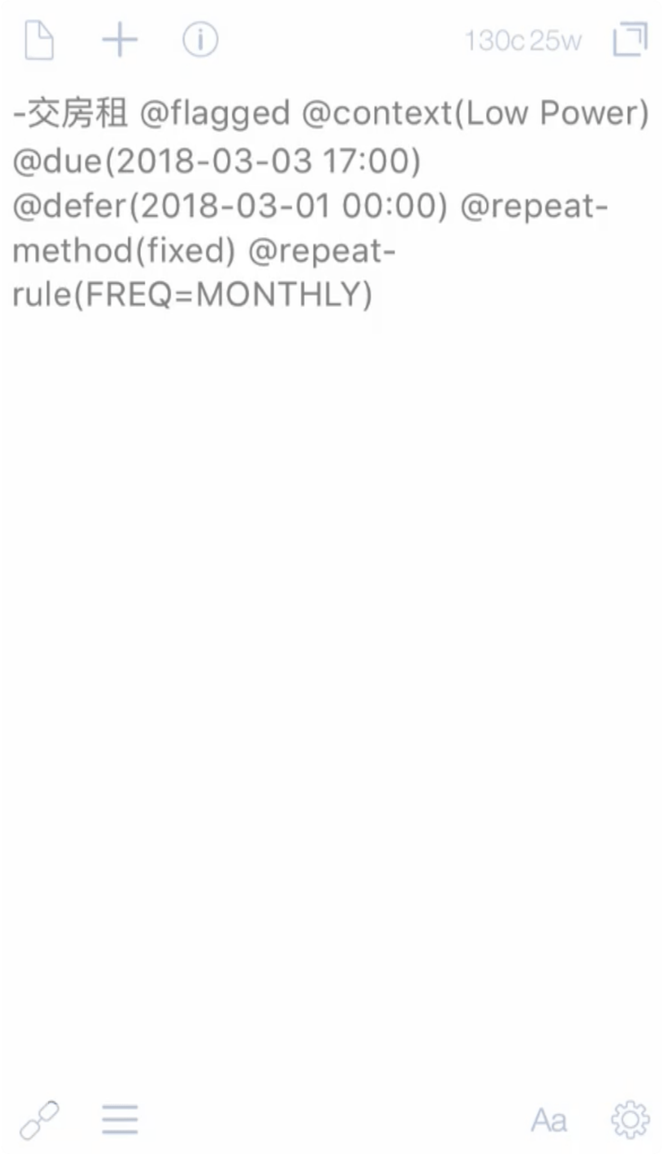
和前面的 TaskPaper 格式比起来，URL Schemes 几乎不是给人来一个个字母输入的（实际上也不是）。2.14 版 OmniFocus 支持导入TaskPaper 格式文本后，我们就可以手写动作内容，按照每个动作的实际需求填写属性来添加动作。OmniFocus 所支持的属性有：

- `@autodone` ：自动完成（true 或者 false）；
- `@context` ：上下文；
- `@defer` ：推迟至日期；
- `@done` ：完成日期；
- `@due` ：截止日期；
- `@estimate` ：持续时间；
- `@flagged` ：标为旗帜；
- `@parallel` ：项目类型为平行（true）或顺序（false）；
- `@repeat-method` ：重复方式， `fixed` 定期重复， `start-after-completion` 完成后再次推迟， `due-after-completion` 完成后再次到期；
- `@repeat-rule` ：重复规则，填写的内容要遵从 ICS 语法，文档见 RFC2445。

有了对 TaskPaper 格式文本的支持，剩下的就是把它导入进 OmniFocus。Drafts 官方创建了一个动作「OmniFocus」（点击下载），就是利用 Taskpaper 格式文本，可以方便地将文本导入进 OmniFocus，实现了我们想要的快速创建动作的效果。仔细查看这个动作的 URL Schme，其实非常简单：

```
omnifocus:///paste?content=[[draft]]&x-success=drafts4://
```

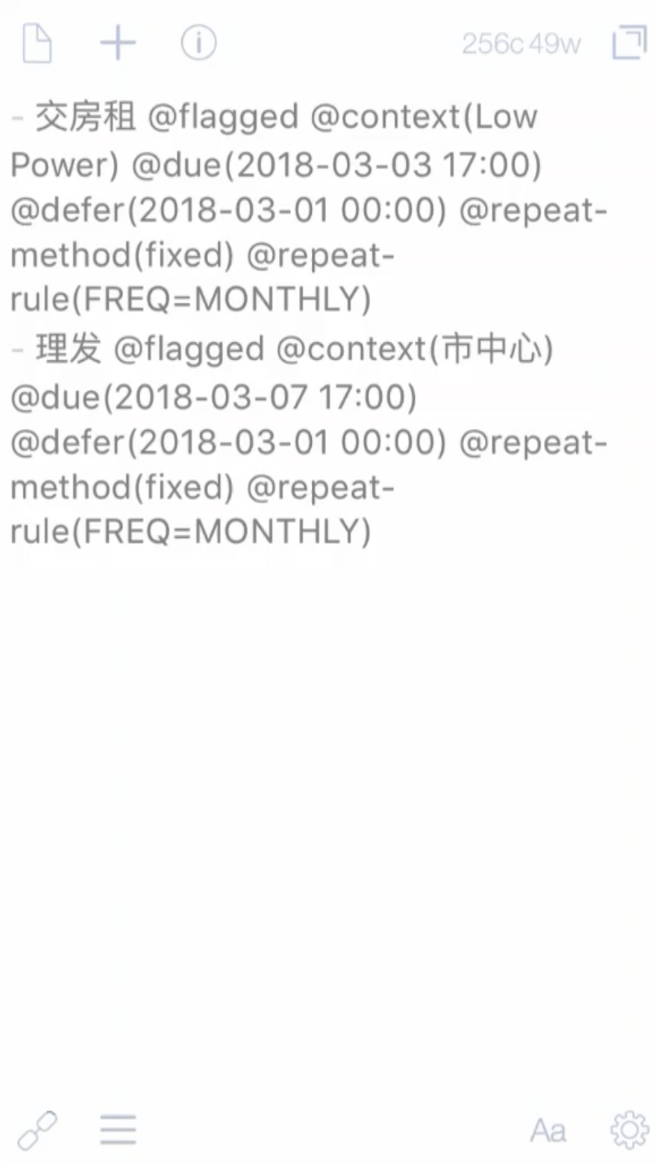
而正好 Drafts 也是一个方便快速打开，记下碎片文字的地方，两者互相协作，不用慢悠悠地打开 OmniFocus 去一个个输入了。



-交房租 @flagged @context(Low Power) @due(2018-03-03 17:00) @defer(2018-03-01 00:00) @repeat-method(fixed) @repeat-rule(FREQ=MONTHLY)

其实这条看似简单的 URL Scheme 暗藏了两个非常让人赞叹的点：

1. 「OmniFocus」这个动作不止能单个添加动作，如果你在 Drafts 中用 Taskpaper 记录了多个动作，它也可以一次性添加到 OmniFocus 的收件箱中。
2. OmniFocus 2 是支持 x-callback-url 的，添加完动 会跳转回 OmniFocus。



- 交房租 @flagged @context(Low Power) @due(2018-03-03 17:00) @defer(2018-03-01 00:00) @repeat-method(fixed) @repeat-rule(FREQ=MONTHLY)
- 理发 @flagged @context(市中心) @due(2018-03-07 17:00) @defer(2018-03-01 00:00) @repeat-method(fixed) @repeat-rule(FREQ=MONTHLY)

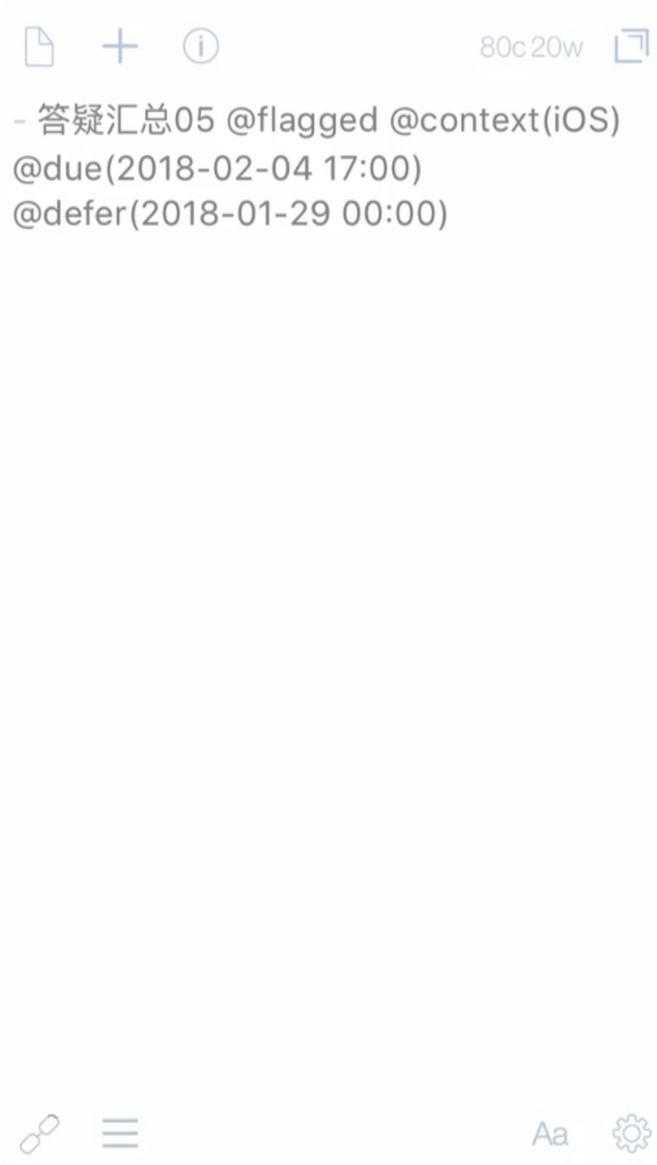
值得注意的是，只有 `/paste` 参数的话，动作都会被添加到收件箱中。如果想添加动作到指定位置，需要加上 `target` ：

- `target=/task/project-name` ：被添加到指定项目
- `target=/task/task-id` ：被添加到指定动作
- `target=/folder/folder-name` ：被添加到指定文件夹

例如我们可以将之前的动作稍加修改：

```
omnifocus:///paste?target=/task/OF&content=[[draft]]
```

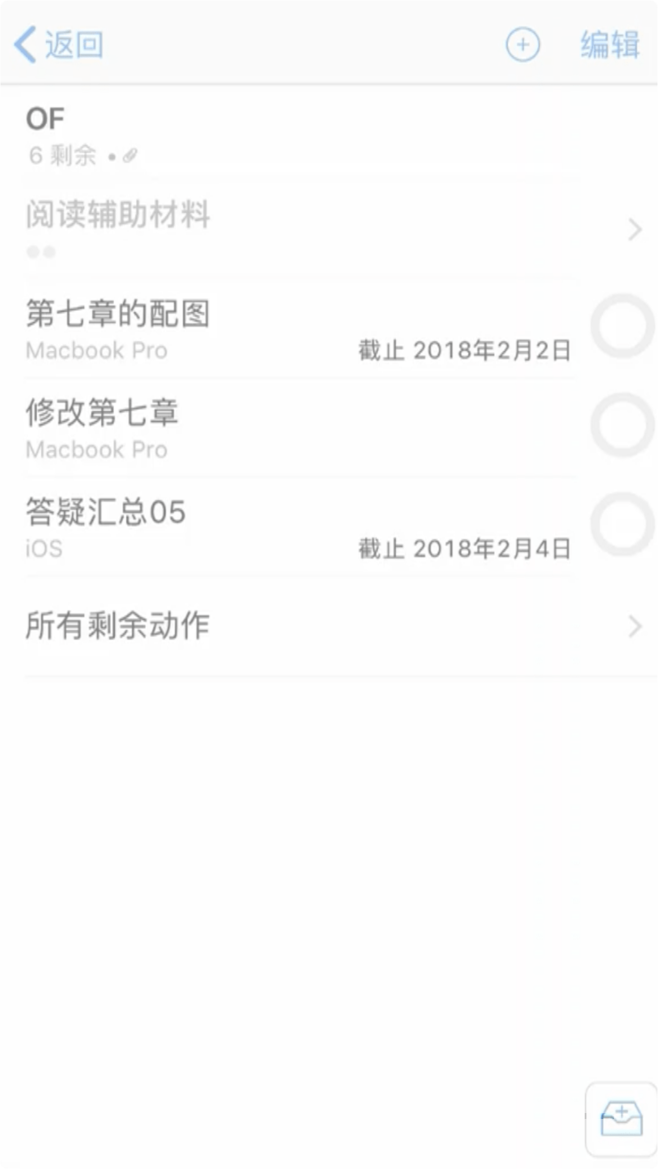
就可以将动作添加到「OF」项目中。



- 答疑汇总05 @flagged @context(iOS) @due(2018-02-04 17:00) @defer(2018-01-29 00:00)

但由于 Drafts 的动作一旦生成，修改起来非常麻烦。如果这次是希望添加到「OF 教程」项目，下次是「生活」文件夹，就得创建多个不同的动作。所以我个人认为直接添加到收件箱中反倒是比较贴合实际的做法。

OmniFocus 既支持 TaskPaper 格式文本的导入，也支持 出。在 Mac 上右键点击动作，就能看到「拷贝为 TaskPaper」的选项。而在 iOS 中，直接拷贝动作即可。



OF
6 剩余

阅读辅助材料

第七章的配图
Macbook Pro　截止 2018年2月2日

修改第七章
Macbook Pro

答疑汇总05
iOS　截止 2018年2月4日

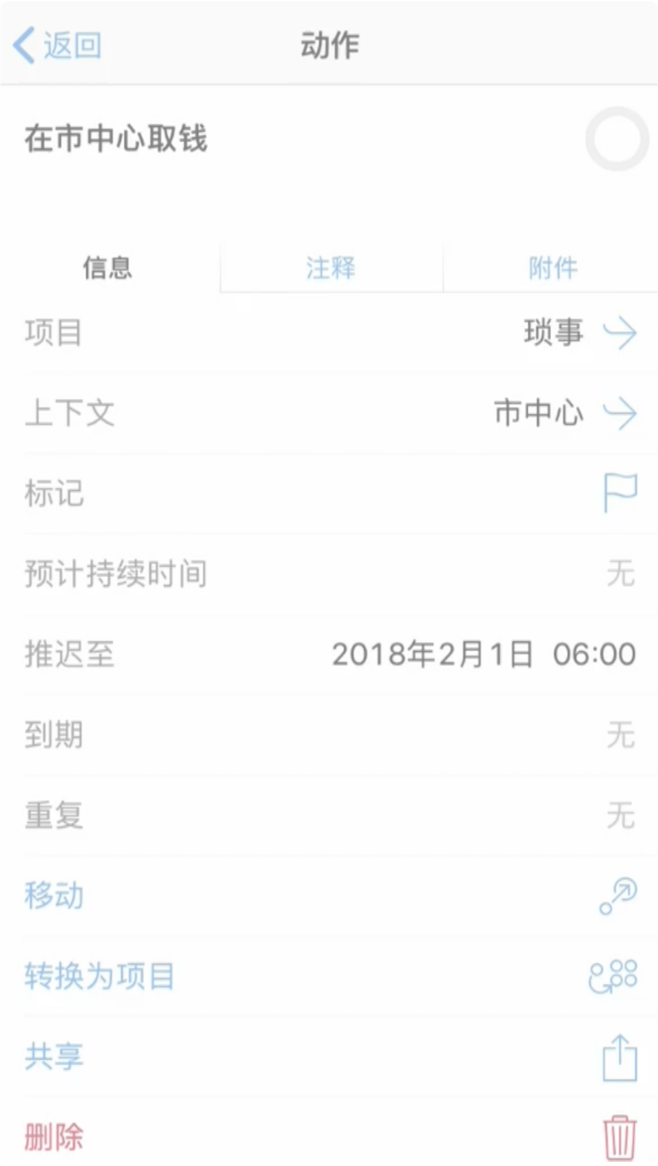
所有剩余动作

和 Workflow 进行协助

Workflow 这款应用的作用类似于 macOS 上的 AppleScript，本身不给应用增加什么功能，但擅长将各个系统功能整合在一起形成一套自动化流程，方便使用者完成一些动起手来非常琐碎的工作。而且不同于之前的 Drafts4、Launch Center Pro 等应用往 OmniFocus 里添加动作，Workflow 是可以把 Omr us 的动作提取出来，转移到其他应用中。

Workflow 和 OmniFocus 结合起来的用法可以说有无限可能，这里仅举一例，以作参考。

在收集任务阶段，我们强调过要收集要快，脑中出现什么就记下什么。但有些有着明确发生时间的动作，其实更适合放在日历中，例如「下午三点去开会」或者「下个月 15 号查信用卡账单」。对于这类任务，需要将它们转移到日历应用中，但如果又纯手动操作，未免太繁琐了。



在市中心取钱

项目 琐事
上下文 市中心
标记
预计持续时间 无
推迟至 2018年2月1日 06:00
到期 无
重复 无
移动
转换为项目
共享
删除

由 Hum 编辑过的 Workflow「OmniFocus 2.14 to Fantastical」这个就可以方便地将 OmniFocus 中的动作转移到 Fantastical 中作为日历项。使用时先在 OmniFocus 中动作页面中通过 Share Sheet 激活 Workflow。如果你给动作设定了推迟至时间和持续时间，就自动作为开始时间和持续时间，否则需要手动选择。运行完成后会在 Fantastical 中创建一个新的项

目，最后跳回 OmniFocus。详细讲解可以阅读 Hum 的《OmniFocus 的 URL Schemes 用法》。

和 Airmail 进行协作

通过 URL Schemes， 给 OmniFocus 添加动作的方式更加丰富了。喜欢 Drafts、Editorial 和 Launch Center Pro 的用户可以直接在这些应用中记录自己的任务，再添加进 OmniFocus。

但说实话这三个应用都偏向 Power User，需要使用者有一定动手能力。所幸第三方开发者对 OmniFocus 的支持程度非常高，一些应用是通过应用内插件形式来与其他应用进行协作，这时一般都会考虑对 OmniFocus 的适配。

Airmail 就是一款支持应用内插件（在 Airmail 中叫做「操作」）的邮件客户端。只要你设备上安装了 OmniFocus，就可以在 Airmail 的 `设置 - 服务` 中先勾选 OmniFocus，再去 `操作` 中给 OmniFocus 打钩，就会在操作列表看到 OmniFocus。

1327346@uni-wuppertal.de

Tscherniewski, Anna 30 1月
[Asim] Schlüssel-Liste
Hallo zusammen, Bitte schickt mir mal alle eure Schlüsselnummern und Berechtigungen (Schl...

和 OmniFocus 提供的 Mail Drop 效果类似，邮件主题作为动作名，内容作为注释。相比 Mail Drop 优势在于，不需要手动发送一份邮件，不会让发件箱显得累赘。

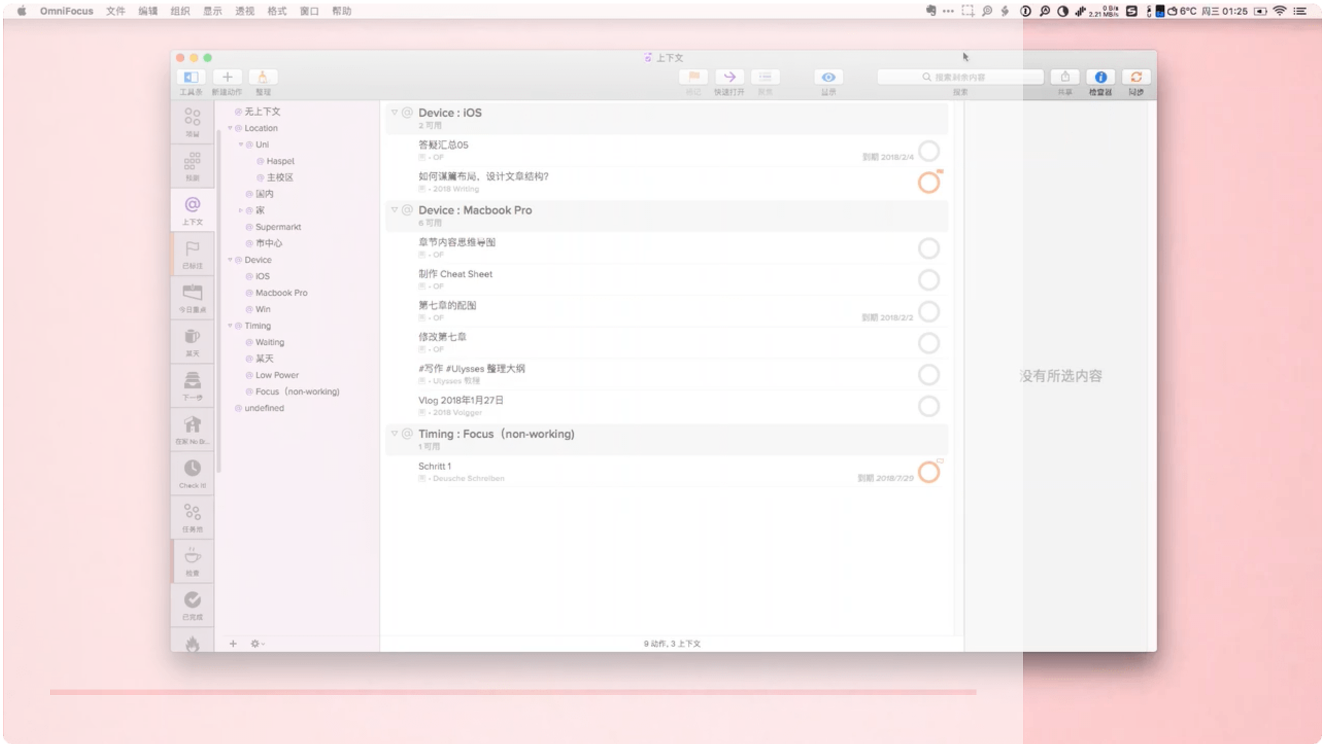
OmniFocus 在 macOS 中与其他应用协作

如果你对第六章讲过的 AppleScript 的内容还印象深刻 会意识到一个事实：理论[1]上借助 AppleScript，OmniFocus 可以和任何一个 Mac 上的应用协作。还记得那个简单到只有七行代码的「Send Safari URL to OmniFocus」 脚本吗？它就实现了 Safari 和 OmniFocus 之间的协作。

既然如此，本节不就像上节讲解那样，去梳理 OmniFocus 与其他应用协作的模式，仅是作为前一章的拾遗补缺，演示一些实用的实例。

和 Evernote 的协作

在上一章中，我们介绍过一个「Weekly project report generator」脚本，自动创建一个 markdown 文本，来保持一周完成的动作和项目。不过很多人都会喜欢在 Evernote 中整理自己的一切文字资料，能不能直接在 Evernote 中生成一篇类似的报告呢？

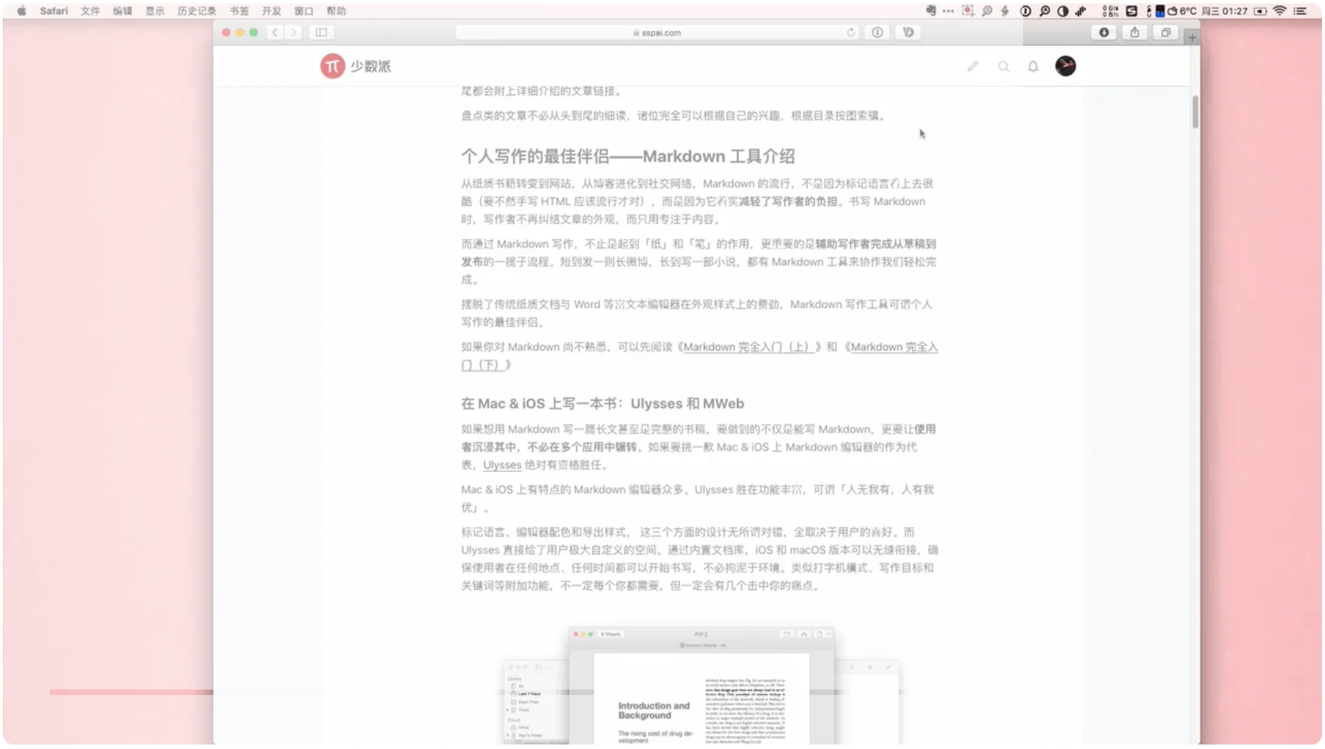

OmniFocus
文件
编辑
组织
显示
透视
格式
窗口
帮助
上下文
工具条
新建动作
整理
快速打开
显示
搜索剩余内容
共享
检查器
同步
无上下文
Location
Uni
Haspel
主校区
国内
家
Supermarkt
市中心
Device
iOS
Macbook Pro
Win
Timing
Waiting
某天
Low Power
Focus（non-working)
undefined
Device : iOS
2 可用
答疑汇总05
OF
到期 2018/2/4
如何谋篇布局，设计文章结构?
2018 Writing
Device : Macbook Pro
6 可用
章节内容思维导图
OF
制作 Cheat Sheet
OF
第七章的配图
OF
到期 2018/2/2
修改第七章
OF
#写作 #Ulysses 整理大纲
Ulysses 教程
Vlog 2018年1月27日
2018 Volgger
Timing : Focus（non-working)
1 可用
Schritt 1
Deusche Schreiben
到期 2018/7/29
没有所选内容
9 动作，3 上下文

Ben Waldie 编写的「end completed task report to Evernote」就是就可以做到同样的效果，甚至比「Wee project report generator」更好用。它不仅能生成本周的任务报告，还可以选择生成今天、昨天、上周、本月的报告。不仅仅是动作标题被记录了，动作相关属性也会被记录。脚本运行完成后，直接生成 Evernote 文本到 Evernote 收件箱中。

和 PopClip 的协作

每个使用电脑的人其实都是「懒惰」的，能不动键盘就只用鼠标点，能少点几下鼠标绝不多点。PopClip 就是这样一款软件，它能让你用鼠标完成一些需要快捷键才能实现的操作。

PopClip 在选中文字后，会自动弹出一排插件选框，让我们只用鼠标就可以完成类似拷贝粘贴、在维基百科中搜索或者查字典等动作。



尾都会附上详细介绍的文章链接。

盘点类的文章不必从头到尾的细读，诸位完全可以根据自己的兴趣，根据目录按图索骥。

个人写作的最佳伴侣——Markdown 工具介绍

从纸质书籍转变到网站，从博客进化到社交网络，Markdown 的流行，不是因为标记语言看上去很酷（要不然手写 HTML 应该流行才对），而是因为它着实减轻了写作者的负担。书写 Markdown 时，写作者不再纠结文章的外观，而只用专注于内容。

而通过 Markdown 写作，不止是起到「纸」和「笔」的作用，更重要的是辅助写作者完成从草稿到发布的一揽子流程。短到发一则长微博，长到写一部小说，都有 Markdown 工具来协作我们轻松完成。

摆脱了传统纸质文档与 Word 等富文本编辑器在外观样式上的费劲，Markdown 写作工具可谓个人写作的最佳伴侣。

如果你对 Markdown 尚不熟悉，可以先阅读《Markdown 完全入门（上）》和《Markdown 完全入门（下）》

在 Mac & iOS 上写一本书：Ulysses 和 MWeb

如果想用 Markdown 写一篇长文甚至是完整的书稿，要做到的不仅是能写 Markdown，更要让使用者沉浸其中，不必在多个应用中辗转。如果要挑一款 Mac & iOS 上 Markdown 编辑器的作为代表，Ulysses 绝对有资格胜任。

Mac & iOS 上有特点的 Markdown 编辑器众多。Ulysses 胜在功能丰富，可谓「人无我有，人有我优」。

标记语言、编辑器配色和导出样式， 这三个方面的设计无所谓对错，全取决于用户的喜好。而 Ulysses 直接给了用户极大自定义的空间。通过内置文档库，iOS 和 macOS 版本可以无缝衔接，确保使用者在任何地点、任何时间都可以开始书写，不必拘泥于环境。类似打字机模式、写作目标和关键词等附加功能，不一定每个你都需要，但一定会有几个击中你的痛点。

PopClip 除了默认动作，还可以上官方拓展库下载插件。在页面下载 OmniFocus 的拓展后，安装后可以使用。PopClip 会将你选择的文字作为动作名和附注。比之前介绍过的通过 OmniFocus 服务来添加动作相比，PopClip 的这个插件还会记录下文字所在的路径或者网页，之后还可以按图索骥，找到原位置搜集更多信息。

小结

关于协作的话题，其实更需要读者自己去尝试。每个人的工具选择都不同，在已有的选择上，能通过 AppleScripts 和 URL Schemes 来将 OmniFocus 与其他应用相互桥接，从而形成一套得心应手的工具流。

从任务的收集整理，透视的创建与回顾，再到 AppleScripts 和应用协作。本章的结束，就意味你已经掌握了关于 OmniFocus 的所有重要内容。

在教程的最后，我希望你已经达到购买时的目的：让 OmniFocus 为你所用。

---

1. 实际上，如果应用不支持 AppleScript，也就没法进行协作了。 ↩

13

上一期
答疑汇总 05

下一期
后记

评论将在经过作者筛选后公开展示

发送

精选评论（3） **我的评论**

泉水沁我心

我刚开始喜欢搞点酷炫的，几个app来回跳，自定义一些弹窗做选择题。现在最想以我个人的语调写个任务，不用@这个那个，自动给任务打上标签，分配到对应的项目。做之前的难点在于分析出相对固定带有指向性的表达方式。做的过程中要设置好if的前后关系和正则表达式匹配，最好形成模块化的，便于以后增加。做完之后在OF设置一个透视，收集未涉及到且可以套用模块的任务类型。我感觉这是一个一劳永逸的做法，不在范围之中的用 Drafts 替补录入。

06月05日

泉水沁我心

早期作者在一篇文章中讲清楚了 推迟至 的真正含义让我豁然开朗，再次表示感谢。第七章看完以后我想我需要坐下来[illegible]分析一下个人的操作习惯和生活工作环境了，傻瓜式的键盘[illegible]留给 Drafts 自定义键盘，个人自然语习惯留给 Workflow 做if的层层过滤，无脑肌肉反应让 Lcp 完成。在 人工智障 成为真正 人工智能 以前，我想没有比这更方便了！

05月31日

sainho 作者

我和你对 Drafts、Workflow 和 LCP 三者的应用场景非常一致！有时候我们会执迷于在一个工具中通过自己的折腾，使得他能完成所有事情。但是或许分场景运用不同的工具，才是既省时间又省脑力的办法！

05月31日

shawnxjf

我想问下，怎么导入taskpaper 到omnifocus 呢。毕竟文本输入是更方便的方式。

05月03日

sainho 作者

不知道你是在 iOS 上还是 Mac 想导入 taskpaper 到 omnifocus？在 iOS 上的相关讲解在第七章，在 Mac 上思路是相同地，不过可能要借助 Alfred 或者 AppleScript，我个人认为是需求不高的（毕竟在电脑上 OF 的界面设计的更合理一些）。这里我找到一个比较老的脚本，不过思路应该是没问题的 http://www.macdrifter.com/2012/02/plain-text-to-omnifocus.html

05月02日

支持我们 作者招募 用户协议 FAQ Contact Us

© 2013-2018 少数派 | 粤ICP备09128966号-4 | CC BY-NC 4.0